

JCS20 U.S. PTO
02/15/00

2-16-00

A

UTILITY PATENT APPLICATION TRANSMITTAL (New Nonprovisional Applications Under 37 CFR § 1.53(b))	Attorney Docket No. <u>1018.070US1</u>
---	---

TO THE ASSISTANT COMMISSIONER FOR PATENTS:

Transmitted herewith is the patent application of () application identifier or (X) first named inventor, Ilan Caron, entitled Automatic Message-Based Module Invocation, for a(n):

(X) Original Patent Application.

() Continuing Application (prior application not abandoned):

() Continuation () Divisional () Continuation-in-part (CIP)

of prior application No: _____ Filed on: _____

() A statement claiming priority under 35 USC § 120 has been added to the specification.

Enclosed are:

(X) Specification; 25 Total Pages.

(X) Drawing(s); 5 Total Sheets.

(X) Oath or Declaration:

(X) A Newly Executed Combined Declaration and Power of Attorney:

() Signed. (X) Unsigned. () Partially Signed.

() A Copy from a Prior Application for Continuation/Divisional (37 CFR § 1.63(d)).

() Incorporation by Reference. The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied, is considered as being part of the disclosure of the accompanying application and is hereby incorporated herein by reference.

() Signed Statement Deleting Inventor(s) Named in the Prior Application. (37 CFR § 163(d)(2)).

() Power of Attorney.

(X) Return Receipt Postcard.

() Associate Power of Attorney.

() A Check in the amount of \$ _____ for the Filing Fee.

() Preliminary Amendment.

() Information Disclosure Statement and Form PTO-1449.

() A Duplicate Copy of this Form for Processing Fee Against Deposit Account.

() A Certified Copy of Priority Documents (if foreign priority is claimed).

() Statement(s) of Status as a Small Entity.

() Statement(s) of Status as a Small Entity Filed in Prior Application, Status Still Proper and Desired.

(X) Data Entry Format sheets for OCR.

CLAIMS AS FILED				
FOR	NO. FILED	NO. EXTRA	RATE	FEE
Total Claims	47	27	\$18.00	\$ 486.00
Independent Claims	4	1	\$78.00	\$ 78.00
Multiple Dependent Claims (if applicable)				\$0.00
Assignment Recording Fee				\$0.00
Basic Filing Fee				\$690.00
Total Filing Fee				\$1,254.00

Charge \$ _____ to Deposit Account _____ pursuant to 37 CFR § 1.25. At any time during the pendency of this application, please charge any fees required or credit any overpayment to this Deposit Account.

Respectfully submitted,

By: Michael A. Dryja
Michael A. Dryja, Attorney of Record, Reg. No. 39662

Date: 2/15/2000

Correspondence Address:

Law Offices of Michael Dryja
704 228th Avenue NE PMB 694
Redmond, WA 98053
Phone: 425-427-5094
Fax: 206-374-2819

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to addressee" service under 37 CFR § 1.10 in an envelope addressed to The Assistant Commissioner for Patents, Washington, DC 20231, on Feb 15, 2000, by Michael Dryja, and having "express mail" mailing label no. EJ243665105US.

Signature of Michael Dryja

JCS25 U.S. PTO
09/504531
02/15/00

09504531 021500

Inventor Information

Inventor One Given Name:: Ilan
Family Name:: Caron
Postal Address Line One:: 1265 23rd Ave E
City:: Seattle
State or Province:: WA
Postal or Zip Code:: 98112
Citizenship Country:: US

Inventor Two Given Name:: Raphael
Family Name:: Renous
Postal Address Line One:: Wallenberg 12
City:: Haifa
Country:: Israel
Postal or Zip Code:: 34990
Citizenship Country:: Israeli

Inventor Three Given Name:: Yifat
Family Name:: Peled
Postal Address Line One:: 48 Hapalmach St.
City:: Haifa
Country:: Israel
Citizenship Country:: Israeli

Inventor Four Given Name:: James
Family Name:: Simpson
Postal Address Line One:: 53 Lower Washington Drive
City:: Bonet Bay
State or Province:: NSW
Country:: Australia
Postal or Zip Code:: 2226
Citizenship Country:: Australian

Correspondence Information

Name Line One:: Michael Dryja, Esq.
Name Line Two:: Law Offices of Michael Dryja
Address Line One:: 704 228th Avenue NE
Address Line Two:: PMB 694
City:: Redmond
State or Province: WA
Postal or Zip Code:: 98053
Telephone:: (425) 427-5094
Fax:: (206) 374-2819
Electronic Mail One:: mike@dryjapat.com
Electronic Mail Two:: sherry@dryja.com

Application Information

Title Line One:: Automatic Message-Based
Title Line Two:: Module Invocation
Total Drawing Sheets:: 5
Formal Drawings?:: Yes
Application Type:: Utility
Docket Number:: 1018.070US1

Representative Information

Registration Number One:: 39,662
Registration Number Two:: 32,628
Registration Number Three:: 32,022

AUTOMATIC MESSAGE-BASED MODULE INVOCATION

FIELD OF THE INVENTION

This invention relates generally to transactional message systems, and more particularly to the invocation of modules such as software components and executable
5 program files triggered by messages within such systems.

BACKGROUND OF THE INVENTION

When organizations need to have large-scale computer systems that hold mission critical information, such as purchase orders, financial information, etc., they usually resort to message transaction systems. Message transaction systems ensure that data is
10 not lost if the system crashes, and also that data is not duplicated – such as ensuring that two copies of the same purchase order are not processed, etc. A transaction is an activity or a request, such as an order, a purchase, a change, or an addition to a database of information. Transactions usually update one or more files on a non-volatile storage such as a hard disk drive, and thus can serve as both an audit trail and a history for future
15 analyses. A transaction can include one or more messages. A transaction is considered committed when all the messages of the transaction have been received and processed.

Frequently, messages are desired to be used to trigger specific functionality, which can be found in modules, which include but are not limited to software components, such as ActiveX components, Java Beans, Component Object Model
20 (COM) components, etc., as known within the art, as well as executable program files – program files ending in .exe, for example. Within the prior art, for messages to trigger a module, an application has to be developed that opens a message queue in which

messages are received, process the messages to check for messages that are desired to trigger the module, and then trigger the module if such messages are found, and dispatch the messages as necessary. However, this is disadvantageous, because to add new functionality triggering, new applications have to be developed, or existing applications have to be redeveloped. This can be expensive, from a time and cost perspective. For this and other reasons, therefore, there is a need for the present invention.

SUMMARY OF THE INVENTION

The present invention relates to automatically invoking modules, such as software components and program executable files, based on messages. In one embodiment, a transactional message system includes one or more queues, a trigger store, and a trigger service. Each queue is capable of receiving messages. The trigger store includes one or more triggers, where each trigger is associated with a queue, and has associated therewith one or more rules, where each rule has a condition and an action. The trigger service is designed to, upon receipt of a message in a queue, check the condition of each rule of each trigger associated with the queue, to determine if the message satisfies the rule's condition. If such satisfaction occurs, then the action of the rule is performed, which can include invocation of a module.

Embodiments of the invention provide for advantages not found within the prior art. For example, rather than forcing development or redevelopment of applications that process messages to invoke modules, embodiments of the invention allow for the writing of rules, and the grouping of the rules in triggers, such that no coding is required. Thus, embodiments of the invention can be said to provide for an end-to-end solution where no coding has to occur. In one embodiment, the system can also include a trigger manager

that is designed to provide for creating, editing, and deleting triggers in a visual, non-programming manner, further easing the ability to automatically invoke modules based on messages.

The invention includes computer-implemented methods, machine-readable media,
5 computerized systems, and computers of varying scopes. Other aspects, embodiments and advantages of the invention, beyond those described here, will become apparent by reading the detailed description and with reference to the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram of an operating environment in conjunction with which
10 embodiments of the invention can be practiced;

FIG. 2 is a diagram showing an overview of a transactional message system in conjunction with which embodiments of the invention can be practiced;

FIG. 3 is a diagram of a transactional message system according to an
embodiment of the invention;

15 FIG. 4 is a diagram showing the relationship of triggers, rules and queues, in accordance with an embodiment of the invention; and,

FIG. 5 is a flowchart of a method according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description of exemplary embodiments of the invention,
20 reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific exemplary embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those

skilled in the art to practice the invention, and it is to be understood that other
embodiments may be utilized and that logical, mechanical, electrical and other changes
may be made without departing from the spirit or scope of the present invention. The
following detailed description is, therefore, not to be taken in a limiting sense, and the
5 scope of the present invention is defined only by the appended claims.

Some portions of the detailed descriptions which follow are presented in terms of
algorithms and symbolic representations of operations on data bits within a computer
memory. These algorithmic descriptions and representations are the means used by those
skilled in the data processing arts to most effectively convey the substance of their work
10 to others skilled in the art. An algorithm is here, and generally, conceived to be a self-
consistent sequence of steps leading to a desired result. The steps are those requiring
physical manipulations of physical quantities. Usually, though not necessarily, these
quantities take the form of electrical or magnetic signals capable of being stored,
transferred, combined, compared, and otherwise manipulated.

15 It has proven convenient at times, principally for reasons of common usage, to
refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the
like. It should be borne in mind, however, that all of these and similar terms are to be
associated with the appropriate physical quantities and are merely convenient labels
applied to these quantities. Unless specifically stated otherwise as apparent from the
20 following discussions, it is appreciated that throughout the present invention, discussions
utilizing terms such as processing or computing or calculating or determining or
displaying or the like, refer to the action and processes of a computer system, or similar
electronic computing device, that manipulates and transforms data represented as

physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

Operating Environment

5 Referring to FIG. 1, a diagram of the hardware and operating environment in conjunction with which embodiments of the invention may be practiced is shown. The description of FIG. 1 is intended to provide a brief, general description of suitable computer hardware and a suitable computing environment in conjunction with which the invention may be implemented. Although not required, the invention is described in the
10 general context of computer-executable instructions, such as program modules, being executed by a computer, such as a personal computer. Generally, program modules include routines, programs, objects, components, data structures, etc., that perform particular tasks or implement particular abstract data types.

Moreover, those skilled in the art will appreciate that the invention may be
15 practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PC's, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a
20 distributed computing environment, program modules may be located in both local and remote memory storage devices.

The exemplary hardware and operating environment of FIG. 1 for implementing the invention includes a general purpose computing device in the form of a computer 20,

including a processing unit 21, a system memory 22, and a system bus 23 that operatively couples various system components include the system memory to the processing unit 21.

There may be only one or there may be more than one processing unit 21, such that the processor of computer 20 comprises a single central-processing unit (CPU), or a plurality

5 of processing units, commonly referred to as a parallel processing environment. The computer 20 may be a conventional computer, a distributed computer, or any other type of computer; the invention is not so limited.

The system bus 23 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a
10 variety of bus architectures. The system memory may also be referred to as simply the memory, and includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic routines that help to transfer information between elements within the computer 20, such as during start-up, is stored in ROM 24. The computer 20 further includes a hard disk drive 27 for reading
15 from and writing to a hard disk, not shown, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD ROM or other optical media.

The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive
20 interface 33, and an optical disk drive interface 34, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer-readable instructions, data structures, program modules and other data for the computer 20. It should be appreciated by those skilled in the art that any type of computer-readable media

which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read only memories (ROMs), and the like, may be used in the exemplary operating environment.

5 A number of program modules may be stored on the hard disk, magnetic disk 29, optical disk 31, ROM 24, or RAM 25, including an operating system 35, one or more application programs 36, other program modules 37, and program data 38. A user may enter commands and information into the personal computer 20 through input devices such as a keyboard 40 and pointing device 42. Other input devices (not shown) may
10 include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port, or a universal serial bus (USB). A monitor 47 or other
15 type of display device is also connected to the system bus 23 via an interface, such as a video adapter 48. In addition to the monitor, computers typically include other peripheral output devices (not shown), such as speakers and printers.

 The computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as remote computer 49. These logical connections are achieved by a communication device coupled to or a part of the
20 computer 20; the invention is not limited to a particular type of communications device. The remote computer 49 may be another computer, a server, a router, a network PC, a client, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 20, although only a memory

storage device 50 has been illustrated in FIG. 1. The logical connections depicted in FIG. 1 include a local-area network (LAN) 51 and a wide-area network (WAN) 52. Such networking environments are commonplace in office networks, enterprise-wide computer networks, intranets and the Internet, which are all types of networks.

When used in a LAN-networking environment, the computer 20 is connected to the local network 51 through a network interface or adapter 53, which is one type of communications device. When used in a WAN-networking environment, the computer 20 typically includes a modem 54, a type of communications device, or any other type of communications device for establishing communications over the wide area network 52, such as the Internet. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program modules depicted relative to the personal computer 20, or portions thereof, may be stored in the remote memory storage device. It is appreciated that the network connections shown are exemplary and other means of and communications devices for establishing a communications link between the computers may be used.

Overview of a Transactional Message System

A simplified overview of a representative transactional message system in conjunction with which embodiments of the invention can be practiced is described in this section of the detailed description, in conjunction with the diagram of FIG. 2, which is a diagram of such a representative system. The system 200 of FIG. 2 includes a sender 202, a receiver 204, and a network 206. Each of the sender and the receiver 204 can in one embodiment be a computer, such as that which has been described in the previous section of the detailed description, in conjunction with FIG. 1, although the invention is

not so limited. The network 206 can be a public network, such as the Internet, or a private network, such as an intranet or a wide-area network (WAN), or a combination public-private network, such as the public Internet connected to an intranet or an extranet. The invention is not so particularly limited.

5 A local transaction is performed on the sender 202, such that all the messages of the transaction are committed to a local messaging system, or none at all. The local messaging system (not shown in FIG. 2), then delivers the messages making up the transaction over the network 206. If any failures occur during this network phase, then the sender 202 is notified, and the sender 202 takes appropriate corrective action.

10 Similarly, a local transaction is on the receiver 204, such that all the messages of the transaction are committed, or none are. It is noted that if the transaction fails on the receiver 204, then the receiver 204 notifies the sender 202 to appropriately recover. Embodiments of the invention are concerned with actions that take place as a result of the receiver 204 receiving predetermined messages, such as messages from a particular
15 sender, relating to a particular subject matter, etc.; the invention is not so limited. This is described in the next section of the detailed description.

Transactional Message System According to an Embodiment of the Invention

 In this section of the detailed description, a transactional message system according to an embodiment of the invention, that provides for automatic invocation of
20 modules, is described, in conjunction with FIG. 3, which is a diagram of such a system. The system 300 of FIG. 3 in one embodiment can correspond to the receiver 204 of FIG. 3 that has been described (e.g., a computer), although the invention itself is not so limited. The system 300 includes a trigger service 302, a trigger store 304, a trigger

manager 306, a trigger database 308, message queues 310a, 310b . . 310n, and modules 312a, 312b . . 312n. It is noted that the invention itself is not limited to a system in which each of the parts of the system 300 as shown in FIG. 3 are present. As a non-limiting example, the trigger manager 306 and/or the trigger database 308 may not be present in a particular embodiment.

Each of the message queues 310 is capable of receiving messages that are part of transactions. The trigger store 304 includes one or more triggers, where each trigger is associated with a queue. Each trigger has associated therewith one or more rules. The trigger store 304 can itself correspond to a particular computer, such that each computer has its own trigger store 304, such that the triggers of the store 304 are references, or pointers, to actual triggers stored in the trigger database 308, which may be global to more than one computer. However, the invention is not limited to such a system architecture, as can be appreciated by those of ordinary skill within the art. Both the trigger store 304 and the trigger database 308 can in one embodiment include data representing triggers, pointers thereto, etc., as stored on a computer-readable medium, such as a hard disk drive, memory, etc. Likewise, each of the messages queues 310 can in one embodiment include data representing messages, as stored on a computer-readable medium.

The trigger service 302 is designed to, upon receipt of a message in one of the queues 310, examine, or check, each rule of each trigger associated with the queue in which the message has been received, to determine if the message satisfies the rule. A rule satisfied by the message may then invoke a module. As shown in FIG. 3, there are modules 312a, 312b . . 312m. Each module 312 can be a software component, an

executable program file, etc.; the invention is not so limited. Thus, as messages arrive in the queues 310, the trigger service 302 examines the rules associated with the queues, and if they are satisfied, can trigger one or more of the modules 312 associated with the rules.

In one embodiment, the trigger service 302 includes a computer program, executed by a processor from a computer-readable medium. In one embodiment, the message itself, or one or more parts thereof, may be passed by the trigger service 302 to the module executed – for example, this is useful where the module is operable on data such as that found within the message. Furthermore, in one embodiment, the trigger service 302 can correspond to a means for checking the condition of each rule for satisfaction by a

message, and performing the action of the rule upon satisfaction thereof.

The trigger manager 306 is designed to provide for creating, editing, and deleting triggers in a visual, non-programming manner. For example, new triggers associated with particular queues can be created, and new rules within those triggers can be created.

Triggers and rules can also be modified (edited) and deleted. Desirably, this is

accomplished within a graphical user interface, such that a user, for example, can accomplish the creating, editing and deleting via standard point, click and drag

functionality as known within the art. Such a graphical user interface thus provides for a non-programming manner by which to create, edit and delete triggers. In one

embodiment, the trigger manager 306 can correspond to a means for creating, editing and

deleting triggers in a visual, non-programming manner.

It is noted that embodiments of the invention are not limited to the manipulation of triggers in a non-programming manner. For example, other triggers can be utilized that are manipulated in a programming manner. Such triggers, for example, can be

specified via a script, in accordance with a scripting language, such that they are manipulated in a non-visual, programming manner, as can be appreciated by those of ordinary skill within the art.

It is noted that the description provided in this section of the detailed description discusses triggers and rules in a general way. In the next section of the detailed description, therefore, triggers and rules according to embodiments of the invention are described more particularly. Such triggers and rules can be utilized in conjunction with the system as described in conjunction with FIG. 3. Furthermore, in another succeeding section of the detailed description, methods according to embodiments of the invention are described. Such methods are amenable to performance by the system as described in conjunction with FIG. 3 as well.

Triggers and Rules

In this section of the detailed description, triggers and rules, in conjunction with which embodiments of the invention, such as the system described in the preceding section, can be utilized, are described in more detail. The description is made with reference to FIG. 4, which is a diagram showing a trigger, a rule, and a queue according to an embodiment of the invention. The diagram of FIG. 4 includes a trigger 400. The trigger 400 is associated with a particular queue 404, and has associated therewith one or more rules, such as the rule 402.

The rule 402 has two parts: a condition 406, and an action 408. The condition 406 of the rule 402 is that part against which incoming messages in the queue 404 are checked for satisfaction. For example, the condition 406 of the rule 402 may specify a particular type of message, a particular sender of the message, particular data contained

within the message, etc.; the invention is not so limited. The action 408 of the rule 402 is that part which is performed upon satisfaction of the condition 406. For example, the action 408 can be the invocation of a particular module or modules, such as software components and executable program files, as has been described.

5 As also shown in FIG. 4, the trigger 400 can include a state 410. In one embodiment, the state 410 is selectable and switchable from an enabled state and a disabled state. When the trigger 400 is in the enabled state, this means that the rules associated therewith have their conditions checked for satisfaction against incoming messages in the queue 404. When the trigger 400 is in the disabled state, this means that
10 the rules associated therewith do not have their conditions checked for satisfaction against incoming message in the queue 404. In effect, the state 410 determines whether the trigger 400 is “on” or “off”.

 In one embodiment, as has been described, there is more than one trigger associated with a queue. In such an embodiment, the triggers may be ordered, such that
15 the rules thereof have their conditions checked against the incoming messages in the queue in that order. Furthermore, in one embodiment, the rules within a particular trigger may be ordered, such that they have their conditions checked against the incoming messages in the queue in that order. Such checking, therefore, is performed in a serial manner. In another embodiment, however, the rules and/or the triggers are unordered,
20 such that the checking of the rules against the incoming messages in the queue is performed in a concurrent manner in a multi-processor and/or multi-threaded operating environment, as known within the art, which lends for increased scalability, as can be appreciated by those of ordinary skill within the art.

Furthermore, in one embodiment, a rule such as the rule 402 of the trigger 400 can be specified as a short-circuit rule. A short-circuit rule is one such that satisfaction of the condition thereof by a message receiving in the queue causes the checking of the satisfaction of the conditions of other rules in that and other triggers to stop. That is, the short-circuit rule “short circuits” the process of checking rules for satisfaction by the message. It is noted that a short-circuit rule only short circuits the checking of those rules that have not been checked yet. Therefore, the use of short-circuit rules is most desirable in an ordered rule and/or an ordered trigger environment.

It is also noted that in one embodiment, a rule can be either destructive or non-destructive. A non-destructive rule, if satisfied by a particular message in the queue, does not cause removal of the message from the queue, such that other rules can be checked against the same message for satisfaction thereof. Conversely, a destructive rule, if satisfied by a particular message in the queue, causes the removal of the message from the queue. This effectively means that other rules that have not yet been checked cannot be checked against this message for satisfaction thereof. In one embodiment, by default, a rule is non-destructive unless otherwise specified.

Methods

In this section, methods of varying embodiments of the invention are described. Such methods can be performed by and in conjunction with systems and computer as described in preceding sections of the detailed description, and can also utilize triggers and rules as described in the preceding section of the detailed description. However, the invention is not so limited. In some embodiments, the methods are computer-implemented. The computer-implemented methods can be realized at least in part as one

or more programs running on a computer – that is, as a program executed from a computer-readable medium such as a memory by a processor of a computer, such as the computer shown in and described in conjunction with FIG. 1. The programs are desirably storable on a machine-readable medium such as a floppy disk or a CD-ROM,
5 for distribution and installation and execution on another computer.

Referring to FIG. 5, a flowchart of a method according to an embodiment is shown. In 500, a message is received in a queue, and is inspected. In 502, for each of one or more triggers associated with the queue, each of one or more rules associated with the trigger is checked for satisfaction by the message. Specifically, the condition of each
10 rule is checked for satisfaction by the message. If, for a particular rule, its condition is satisfied, then the method proceeds to 503. If the rule is destructive, as has been described, then the method proceeds from 503 to 505, where the message is removed from the queue, and the method proceeds thereafter to 504. Otherwise, the message is maintained in the queue, and the method proceeds directly to 504 from 503. In 504, the
15 action associated with the rule having its condition satisfied is performed, such as activating modules associated therewith, as has been described in specific detail in previous sections of the detailed description. If, for a particular rule, its condition is not satisfied, then the method proceeds to 506, where the method is finished as to that rule. As described in the previous section of the detailed description, the rules may be checked
20 in a serial or a concurrent manner, may be destructive or non-destructive, may be short-circuit rules, etc.

Conclusion

Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the present invention. Therefore, it is manifestly intended that this invention be limited only by the claims and equivalents thereof.

005720 "TEST" 050560

We claim:

1. A computer-implemented method performable within a transactional message system comprising:

receiving a message in a queue;

5 for each of at least one trigger associated with the queue, checking whether any of at least one rule associated with the trigger has a condition thereof satisfied by the message; and,

upon determining that any of the at least one rule associated with the trigger has a condition thereof satisfied by the message, for each rule having a condition satisfied by
10 the message, performing an action associated with the rule.

2. The method of claim 1, wherein the at least one trigger comprises a plurality of ordered triggers.

3. The method of claim 1, wherein performing an action associated with the rule comprises activating each of at least one module associated with the rule.

15 4. The method of claim 3, wherein each module comprises one of: a software component; and, an executable program file.

5. The method of claim 3, wherein activating each of at least one module associated with the rule comprises passing the message to the module.

6. The method of claim 1, where each trigger has an enabled state and a disabled state, such that the condition of each of the at least one rule of a trigger is checked for satisfaction by the message received in the queue only when the trigger is in the enabled state.

5 7. The method of claim 1, wherein each of at least one of the rules comprises a short-circuit rule, such that satisfaction by the message received in the queue of the condition of the rule causes checking for satisfaction of the condition of each of any non-checked rules to stop.

8. The method of claim 1, wherein each of at least one of the rules comprises a
10 destructive rule, such that satisfaction by the message received in the queue of the condition of the rule removes the message from the queue.

9. The method of claim 1, wherein checking is performed in a serial manner.

10. The method of claim 1, wherein checking is performed in a concurrent manner.

11. A machine-readable medium having instructions stored thereon for execution by a
15 processor to perform a method comprising:

receiving a message in a queue;

checking whether a condition of a rule associated with a trigger associated with the queue is satisfied by the message; and,

upon determining that the condition of the rule is satisfied by the message,
performing an action associated with the rule.

12. The medium of claim 11, wherein performing an action associated with the rule
comprises activating each of at least one module associated with the rule.

5 13. The medium of claim 12, wherein each module comprises one of: a software
component; and, an executable program file.

14. The medium of claim 12, wherein activating each of at least one module associated
with the rule comprises passing the message to the module.

15 15. The medium of claim 11, where the trigger has an enabled state and a disabled state,
such that the condition of each of the rule associated with the trigger is checked for
satisfaction by the message received in the queue only when the trigger is in the enabled
state.

16. The medium of claim 11, wherein the rule comprises a short-circuit rule, such that
satisfaction by the message received in the queue of the condition of the rule causes
15 checking for satisfaction of the condition of any non-checked rules to stop.

17. The medium of claim 11, wherein the rule comprises a destructive rule, such that
satisfaction by the message received in the queue of the condition of the rule removes the
message from the queue.

18. The medium of claim 11, wherein checking is performed in a serial manner.

19. The medium of claim 11, wherein checking is performed in a concurrent manner.

20. A transactional message system comprising:

at least one queue, each queue capable of receiving a plurality of messages;

5 a trigger store of at least one trigger, each trigger associated with a queue, having a state selected from one of an enabled state and a disabled state, and having associated therewith at least one rule, each rule having a condition and an action; and,

10 a trigger service designed to, upon receipt of a message in a queue, check the condition of each rule of each trigger associated with the queue that is in the enabled state for satisfaction by the message, such that the action of the rule is performed upon satisfaction of the condition of the rule by the message.

21. The system of claim 20, wherein the trigger store corresponds to a particular computer and references each of the at least one trigger within a trigger database.

15 22. The system of claim 20, wherein each of the at least one queue comprises data stored on a computer-readable medium.

23. The system of claim 20, wherein each of the at least one trigger store comprises data stored on a computer-readable medium.

24. The system of claim 20, wherein the trigger service comprises a computer program executed by a processor from a computer-readable medium.

25. The system of claim 20, further comprising a trigger manager designed to provide for creating, editing and deleting triggers in a visual, non-programming manner.

5 26. The system of claim 20, wherein the trigger store of the at least one trigger comprises a trigger store of a plurality of ordered triggers.

27. The system of claim 20, wherein the trigger service is further designed to perform the action associated with a rule by activating each of at least one module associated with the rule.

10 28. The system of claim 27, further comprising at least one module, such that the at least one module associated with the rule as activated by the trigger service are selected from the at least one module.

29. The system of claim 28, wherein each module comprises one of: a software component, and an executable program file.

15 30. The system of claim 28, wherein the trigger service is further designed to activate each of the at least one module associated with the rule such that the message to the module is passed to the module.

31. The system of claim 20, wherein each of at least one of the rules comprises a short-circuit rule, such that satisfaction by the message received in the queue of the condition of the rule causes the trigger service to stop checking for satisfaction of the condition of each of any non-checked rules.

5 32. The system of claim 20, wherein each of at least one of the rules comprises a destructive rule, such that satisfaction by the message received in the queue of the condition of the rule removes the message from the queue.

33. The system of claim 20, wherein checking is performed in a serial manner.

34. The system of claim 20, wherein checking is performed in a concurrent manner.

10 35. The system of claim 20, wherein the system comprises at least one computer.

36. A computer for inclusion into a transactional message system comprising:

at least one queue, each queue capable of receiving a plurality of messages;

a trigger store of at least one trigger, each trigger associated with a queue, having a state selected from one of an enabled state and a disabled state, and having associated

15 therewith at least one rule, each rule having a condition and an action; and,

means for, upon receipt of a message in a queue, checking the condition of each rule of each trigger associated with the queue that is in the enabled state for satisfaction by the message, and performing the action of the rule upon satisfaction of the condition of the rule by the message.

37. The computer of claim 36, wherein the trigger store references each of the at least one trigger within a trigger database.

38. The computer of claim 36, further comprising means for creating, editing and deleting triggers in a visual, non-programming manner.

5 39. The computer of claim 36, wherein the trigger store of the at least one trigger comprises a trigger store of a plurality of ordered triggers.

40. The computer of claim 36, wherein the means for checking and performing is further for performing the action associated with a rule by activating each of at least one module associated with the rule.

10 41. The computer of claim 40, further comprising at least one module, such that the at least one module associated with the rule service are selected from the at least one module.

42. The computer of claim 40, wherein each module comprises one of: a software component, and an executable program file.

15 43. The computer of claim 40, wherein the means for checking and performing is further for activating each of the at least one module associated with the rule such that the message to the module is passed to the module.

44. The computer of claim 36, wherein each of at least one of the rules comprises a short-circuit rule, such that satisfaction by the message received in the queue of the condition of the rule causes the means for checking and performing to stop checking for satisfaction of the condition of each of any non-checked rules.

5 45. The computer of claim 36, wherein each of at least one of the rules comprises a destructive rule, such that satisfaction by the message received in the queue of the condition of the rule removes the message from the queue.

46. The computer of claim 36, wherein checking is performed in a serial manner.

47. The computer of claim 36, wherein checking is performed in a concurrent manner.

10

ABSTRACT OF THE DISCLOSURE

Automatic message-based module invocation within a transactional message system is described. The modules can be software components, executable program files, etc. In one embodiment, a system includes one or more queues, a trigger store, and a trigger service. Each queue is capable of receiving messages. The trigger store includes one or more triggers, where each trigger is associated with a queue, and has associated therewith one or more rules, where each rule has a condition and an action. The trigger service is designed to, upon receipt of a message in a queue, check the condition of each rule of each trigger associated with the queue, to determine if the message satisfies the rule's condition. If such satisfaction occurs, then the action of the rule is performed, which can include invocation of a module. The rules may be short-circuit rules, non-destructive rules, and/or destructive rules, and may be checked in either a concurrent or serial manner.

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to addressee" service under 37 CFR § 1.10 in an envelope addressed to The Assistant Commissioner for Patents, Washington, DC 20231, on Feb 15, 2000, by Michael Dryja, and having "express mail" mailing label no. EJ243665105US.

Signature of Michael Dryja

FIG. 1

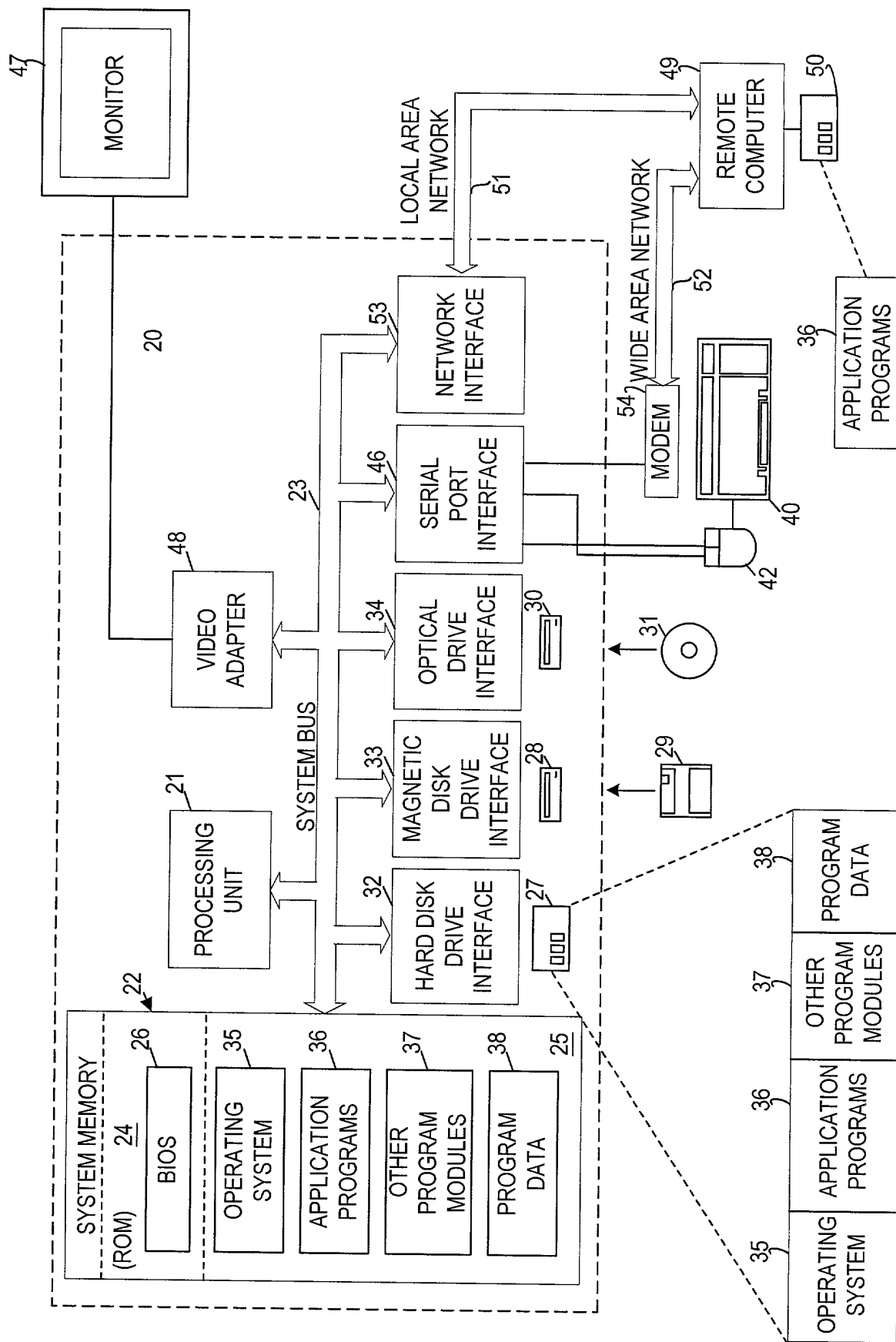


FIG 2

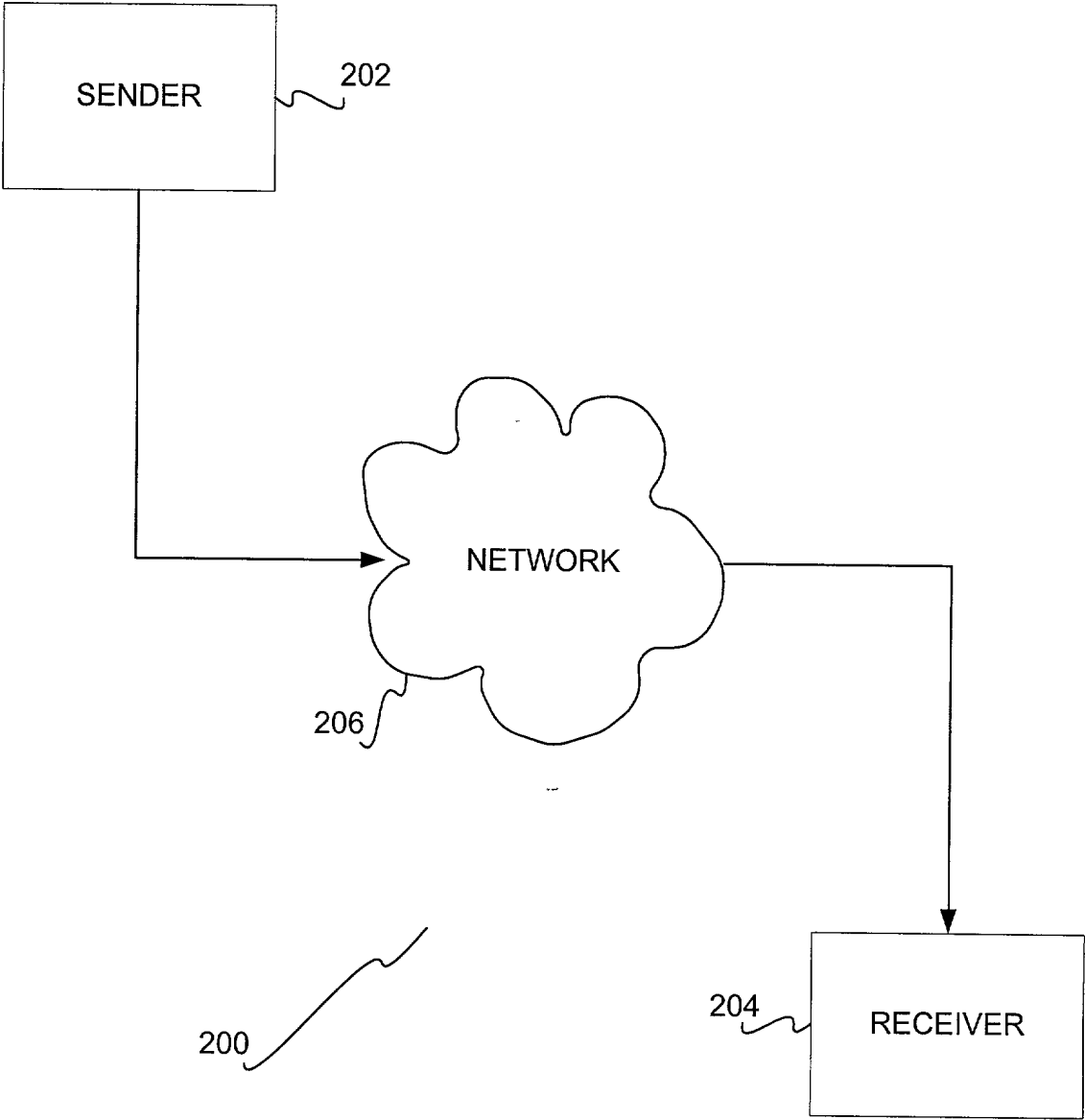


FIG 3

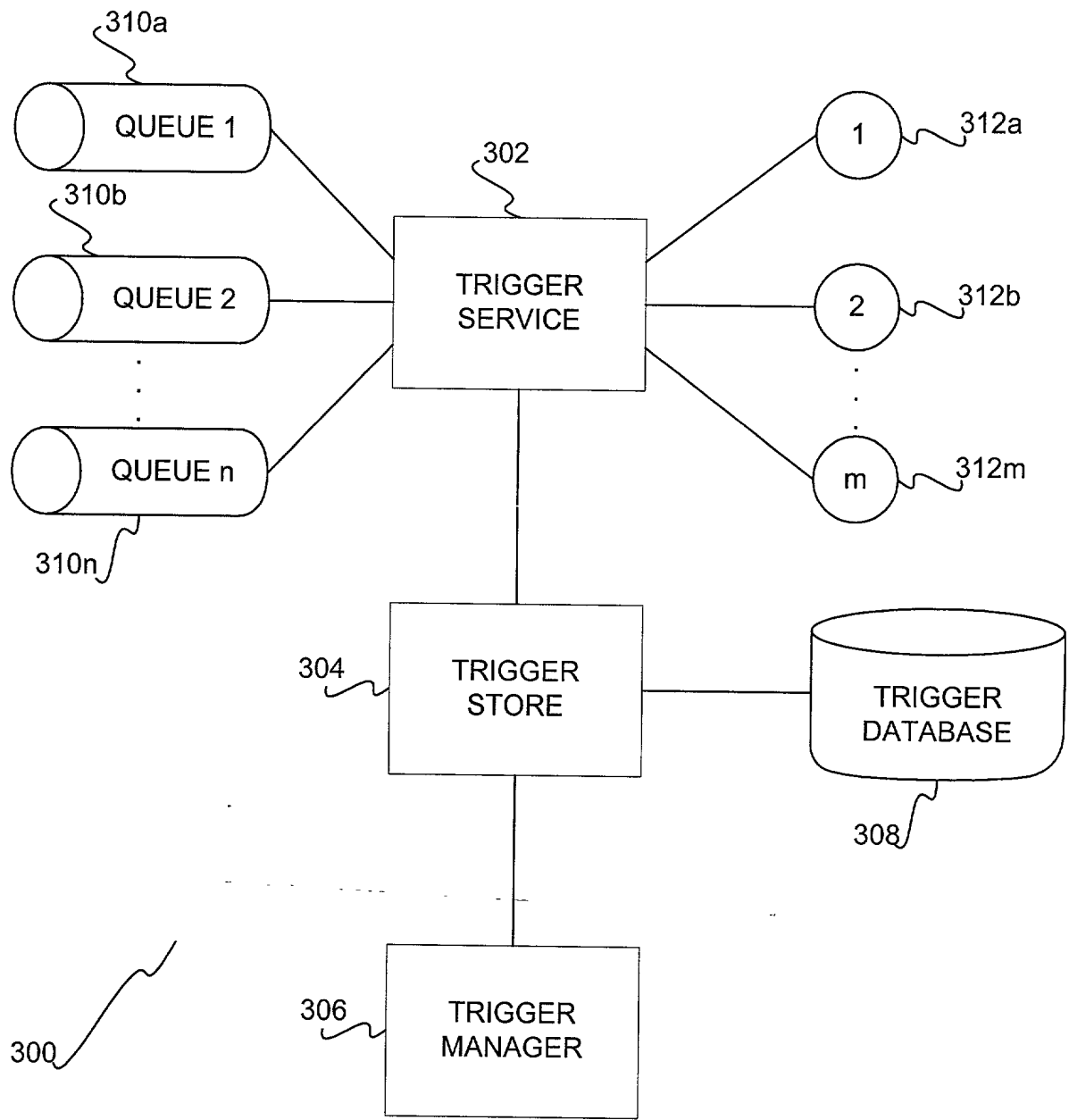


FIG 4

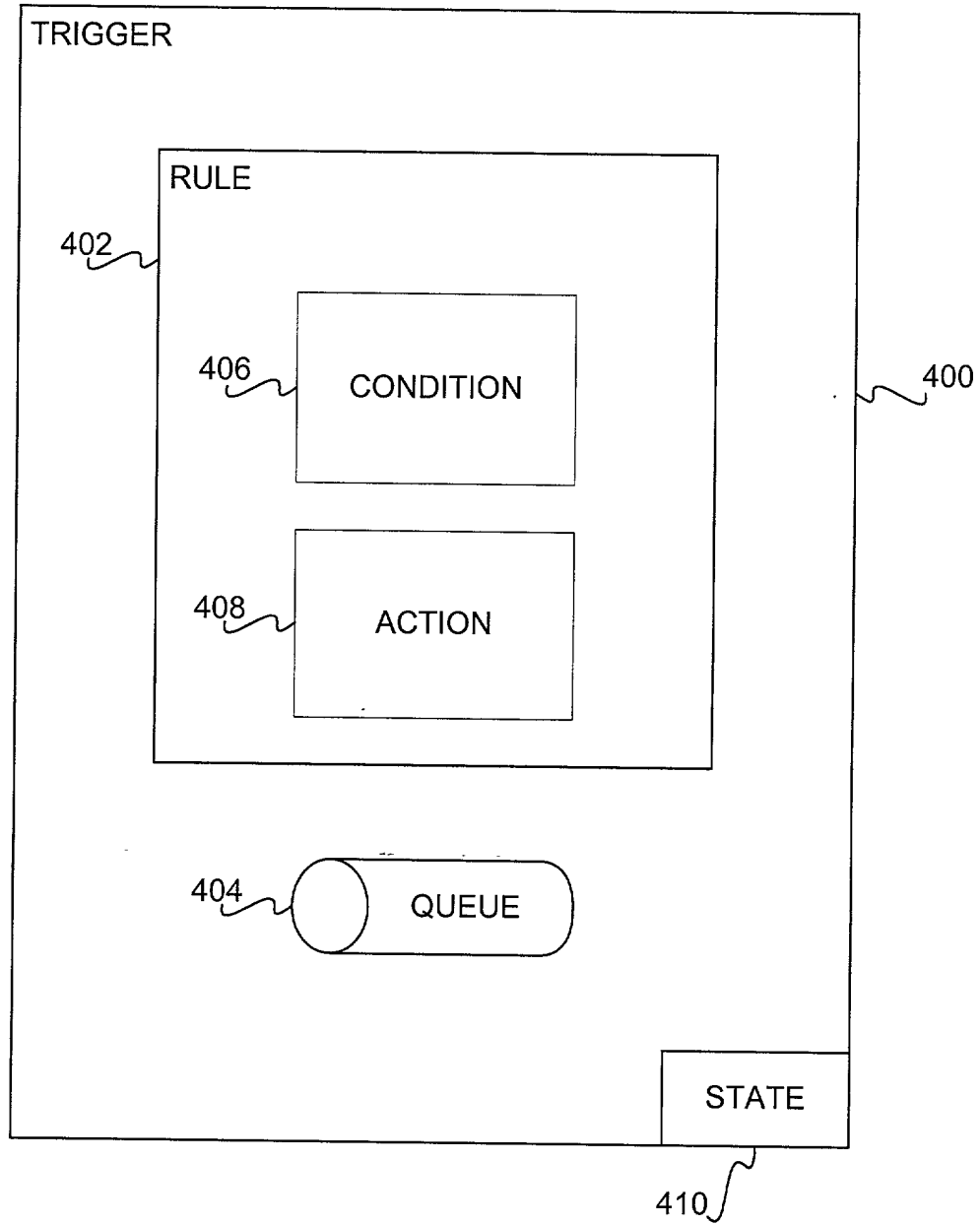
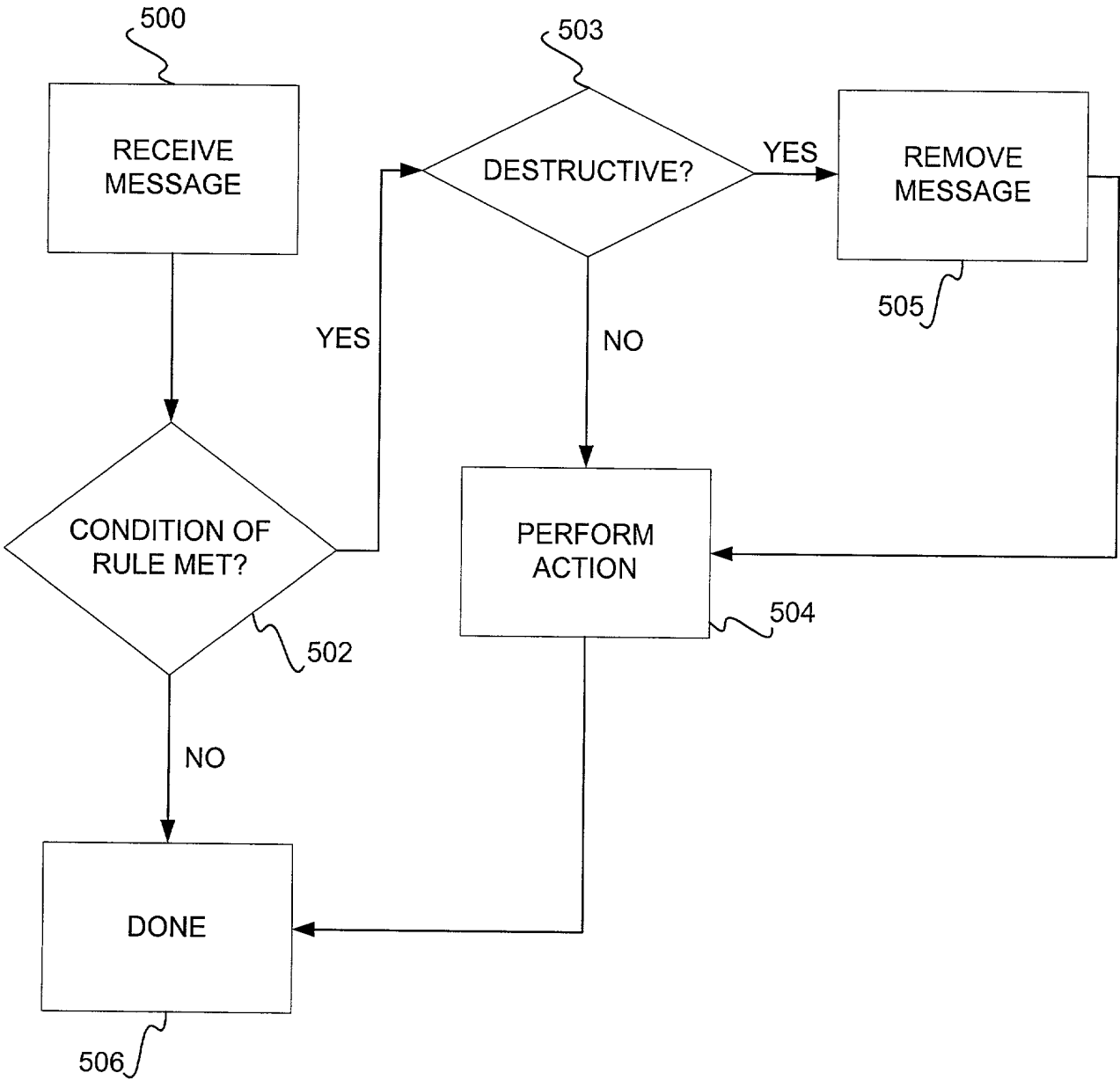


FIG 5



PATENT APPLICATION

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

ATTORNEY DOCKET NO. 1018.070US1

As a below named inventor, I hereby declare that:

My residence/post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

Automatic Message-Based Module Invocation

the specification of which is attached hereto unless the following box is checked:

☐ was filed on _____ as US Application Serial No. or PCT International Application
Number _____ and was amended on _____ (if applicable).

I hereby state that I have reviewed and understood the contents of the above-identified specification, including the claims, as amended by any amendment(s) referred to above. I acknowledge the duty to disclose all information which is material to patentability as defined in 37 CFR 1.56.

Foreign Application(s) and/or Claim of Foreign Priority

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor(s) certificate listed below and have also identified below any foreign application for patent or inventor(s) certificate having a filing date before that of the application on which priority is claimed:

COUNTRY	APPLICATION NUMBER	DATE FILED	PRIORITY CLAIMED UNDER 35 U.S.C. 119
			YES: _____ NO: _____
			YES: _____ NO: _____

Provisional Application

I hereby claim the benefit under Title 35, United States Code Section 119(e) of any United States provisional application(s) listed below:

APPLICATION SERIAL NUMBER	FILING DATE

U.S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION SERIAL NUMBER	FILING DATE	STATUS(patented/pending/abandoned)

POWER OF ATTORNEY:

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) listed below to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

Michael A. Dryja, Reg. No. 39662

Katie E. Sako, Reg. No. 32628

Daniel D. Crouse, Reg. No. 32022

Send Correspondence to:

Michael A. Dryja
Law Offices of Michael Dryja
704 228th Avenue NE
PMB 694
Redmond, WA 98053

Direct Telephone Calls To:

Michael A. Dryja
425-427-5094

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon

Full Name of Inventor: Ilan Caron

Citizenship: US

Residence: 1265 23rd Ave E, Seattle, WA 98112

Post Office Address: Same

Inventor's Signature _____

Date _____

**DECLARATION AND POWER OF ATTORNEY
FOR PATENT APPLICATION (continued)**

ATTORNEY DOCKET NO. 1018.070US1

Full Name of Inventor: Raphael Renous

Citizenship: Israeli

Residence: Wallenberg 12, Haifa, 34990 Israel

Post Office Address: Same

Inventor's Signature

Date

Full Name of Inventor: Yifat Peled

Citizenship: Israeli

Residence: 48 Hapalmach St., Haifa, Israel

Post Office Address: Same

Inventor's Signature

Date

Full Name of Inventor: James Simpson

Citizenship: Australian

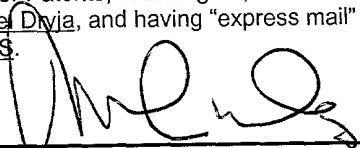
Residence: 53 Lower Washington Drive, Bonet Bay, NSW 2226 Australia

Post Office Address: Same

Inventor's Signature

Date

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to addressee" service under 37 CFR § 1.10 in an envelope addressed to The Assistant Commissioner for Patents, Washington, DC 20231, on Feb 15, 2000, by Michael Dryja, and having "express mail" mailing label no. EJ243665105US.



Signature of Michael Dryja